

COP 2337C Lab 3  
25 points

**BankAccount Inheritance**

- a) From the attached abstract **BankAccount** class, derive a class called **Savings**.

The class should contain:

- (1) A private data member called **interestRate** of type double.
- (1) A default constructor.
- (1) A public setter method, named **SetInterestRate()**, that makes it possible to assign an interest rate to the private data member **interestRate** of the class.
- (1) A public getter method, named **GetInterestRate()**, that is used to get the value of the private data member **interestRate**.
- (2) Override the method **MonthlyFees()**.  
This method should return a monthly fee of \$5.00.
- (2) Override the **DisplayFees()** method from the super class to display the following info: account number, interest rate, and monthly fees.  
*Note: Make calls to the super class methods whenever possible.*

- b) From the attached abstract **BankAccount** class, derive a class called **Checking**.

The class should contain:

- (1) A private data member called **numChecksWritten** of type integer.
- (1) A default constructor.
- (1) A public method **void WriteCheck()** that adds one to the number of checks written.
- (2) A public method **void NewMonth()** that resets the number of checks written to zero.
- (2) Override the method **MonthlyFees()**.  
This method should return a monthly fee of \$10.00 plus a fee of \$.10 for every check written.
- (2) Override the **DisplayFees()** method from the super class to display the following info: account number, number of checks written, and monthly fees.  
*Note: Make calls to the super class methods whenever possible.*

- c) (4 pts) Create a UML diagram to represent the BankAccount Hierarchy.

- d) (4 pts) Create a driver class called **BankAccountMain** to fully test your classes.